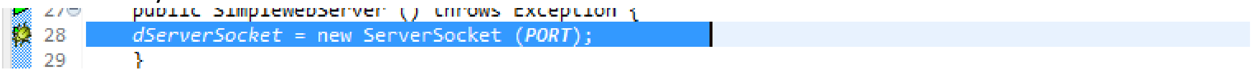
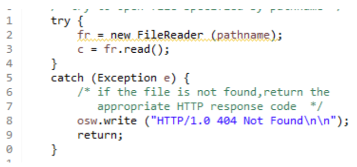
Part 1:

Manual Analysis:



```
    public SimpleWebServer () throws Exception {
        dServerSocket = new ServerSocket (PORT);
    }
```

Final can be added in front of the code to clear the warning, this warning is found by eclipse before execution of the program.



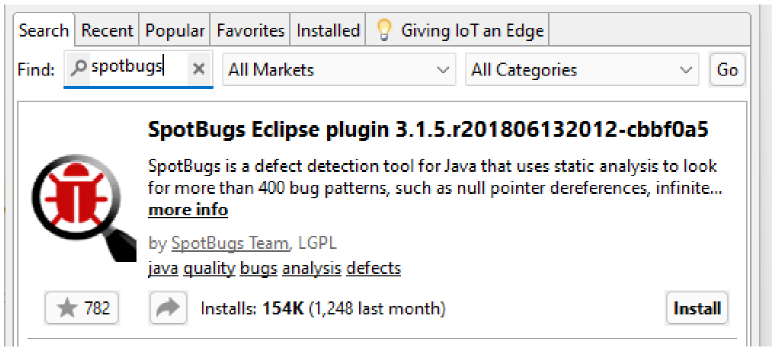
```
    try {
        fr = new FileReader (pathname);
        c = fr.read();
    }
    catch (Exception e) {
        /* if the file is not found,return the
           appropriate HTTP response code  */
        osw.write ("HTTP/1.0 404 Not Found\n\n");
        return;
    }
```

Included: FileReader that is declared and opened here is not closed in the program. So we need to add fr.close() at the end of the code.

Legitimate finding: the prgm code which is inside of the main class is not written in the try-catch block. A character array usage would be better then strings used in the program which can decrease program vulnerability.

Tool Choices/Versions :

- Windows 11
- SporBugs eclipser plugin



SpotBugs Eclipse plugin 3.1.5.r201806132012-cbbf0a5

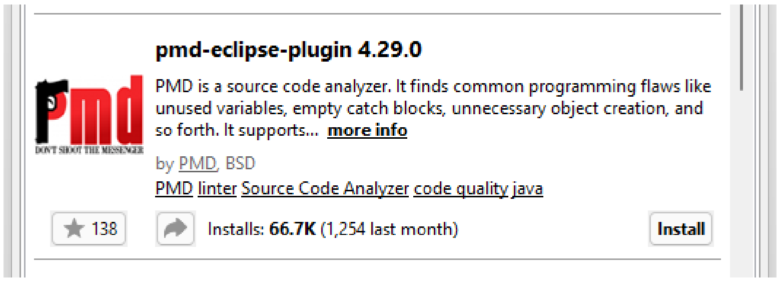
SpotBugs is a defect detection tool for Java that uses static analysis to look for more than 400 bug patterns, such as null pointer dereferences, infinite... more info

by SpotBugs Team, LGPL

java quality bugs analysis defects

Installs: 154K (1,248 last month)

- Pmd eclipse pluguin



pmd-eclipse-plugin 4.29.0

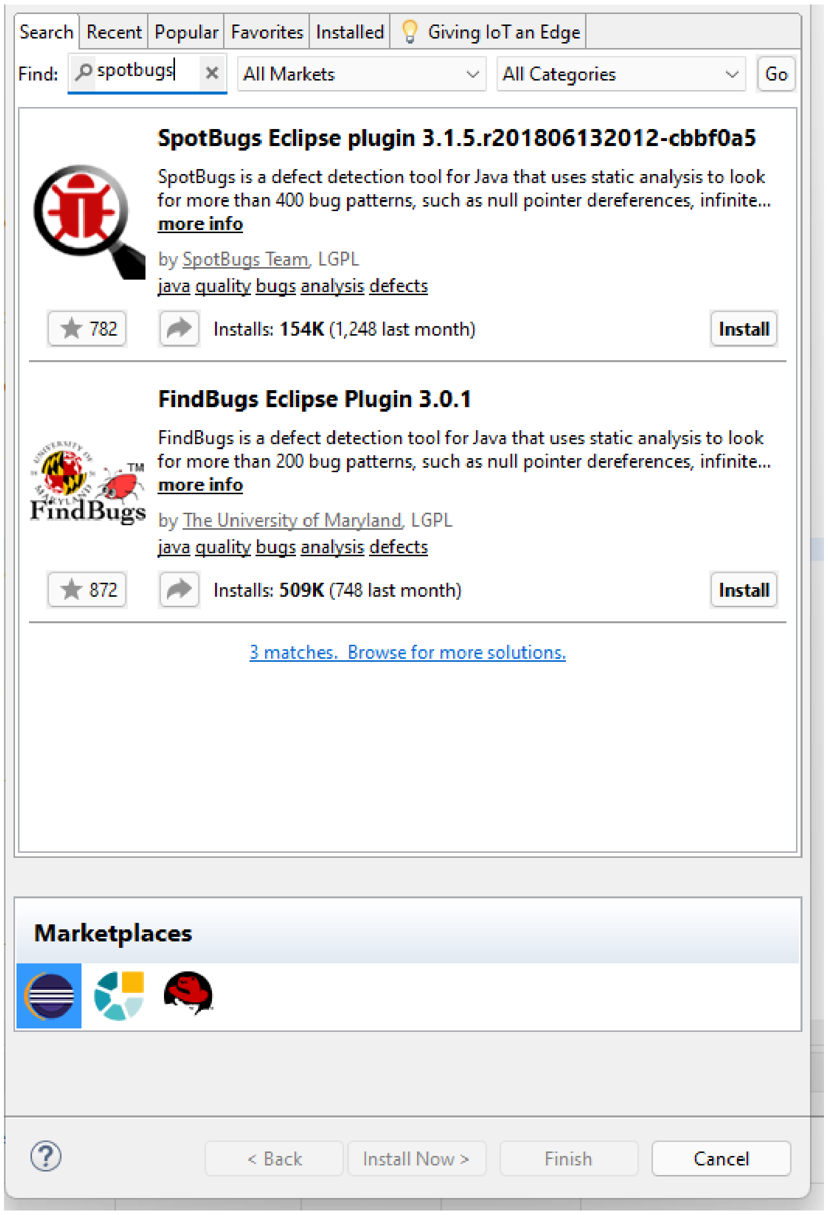
PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It supports... more info

by PMD, BSD

PMD linter Source Code Analyzer code quality java

Installs: 66.7K (1,254 last month)

Tool Invocation Process:
We use the latest version of eclipse
The tools/ plugins are installed from the eclipse market place. We search for spotbugs and click on install and the ide prompts us to restart and click on yes.



SpotBugs Eclipse plugin 3.1.5.r201806132012-cbbf0a5

SpotBugs is a defect detection tool for Java that uses static analysis to look for more than 400 bug patterns, such as null pointer dereferences, infinite... more info

by SpotBugs Team, LGPL

java quality bugs analysis defects

782 Installs: **154K** (1,248 last month) Install

FindBugs Eclipse Plugin 3.0.1

FindBugs is a defect detection tool for Java that uses static analysis to look for more than 200 bug patterns, such as null pointer dereferences, infinite... more info

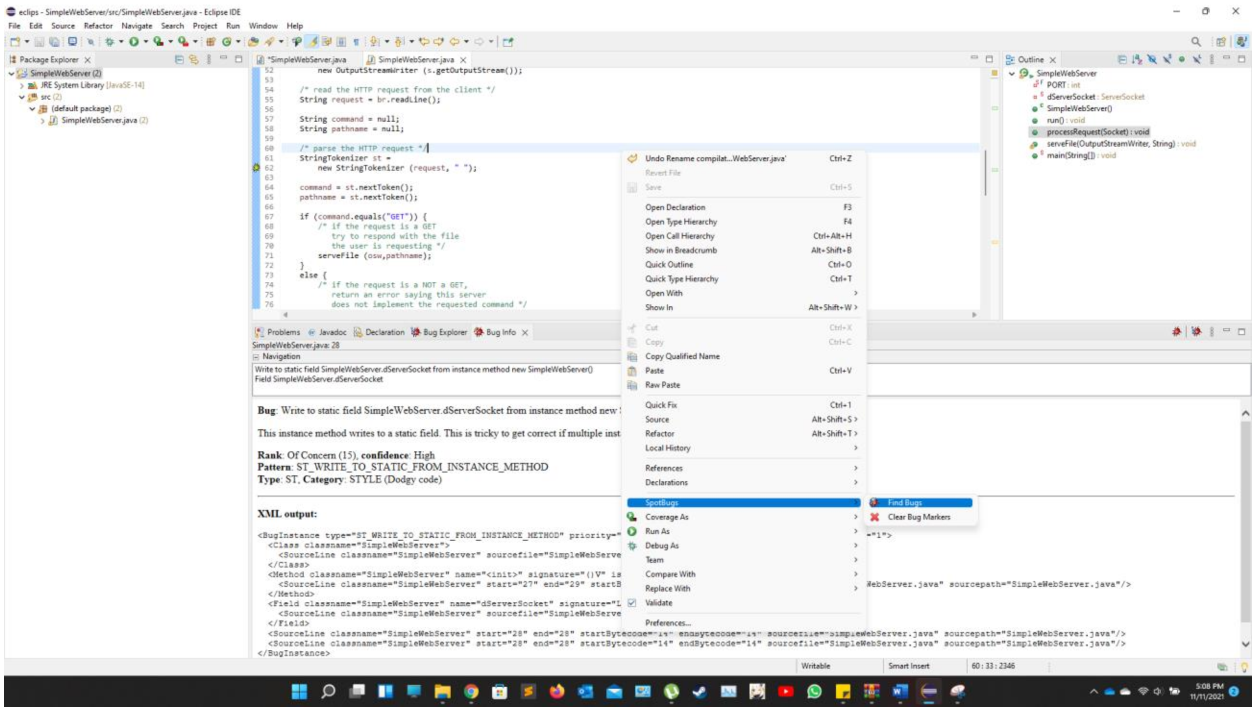
by The University of Maryland, LGPL

java quality bugs analysis defects

872 Installs: **509K** (748 last month) Install

3 matches. Browse for more solutions.

Once the plugin is installed you can right click and then go to spotbugs and click on find bugs.



```
            new OutputStreamWriter (s.getOutputStream());

        /* read the HTTP request from the client */
        String request = br.readLine();

        String command = null;
        String pathname = null;

        /* parse the HTTP request */
        StringTokenizer st =
            new StringTokenizer (request, " ");

        command = st.nextToken();
        pathname = st.nextToken();

        if (command.equals("GET")) {
            /* if the request is a GET
               try to respond with the file
               the user is requesting */
            serveFile (osw,pathname);
        }
        else {
            /* if the request is a NOT a GET,
               return an error saying this server
               does not implement the requested command */
```

SimpleWebServer.java: 28

Navigation

Write to static field SimpleWebServer.dServerSocket from instance method new SimpleWebServer()
Field SimpleWebServer.dServerSocket

Bug: Write to static field SimpleWebServer.dServerSocket from instance method new

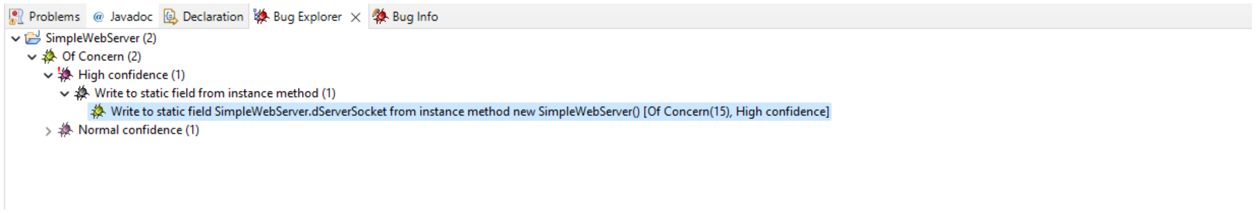
This instance method writes to a static field. This is tricky to get correct if multiple inst

Rank: Of Concern (15), **confidence**: High
Pattern: ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD
Type: ST, **Category**: STYLE (Dodgy code)

XML output:

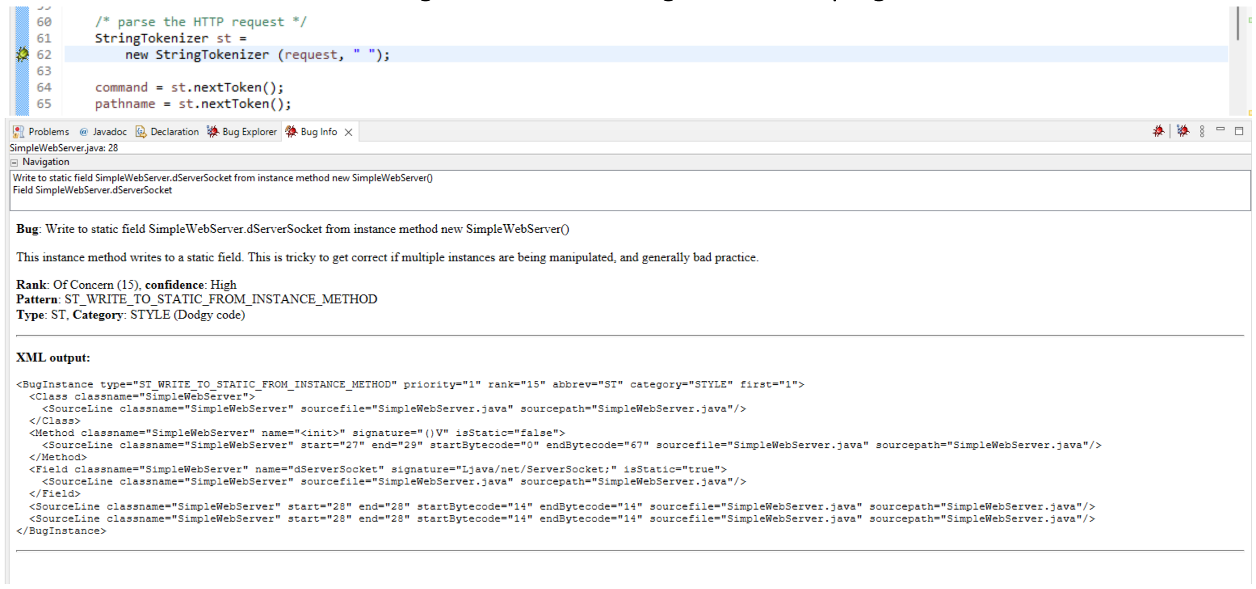
```
<BugInstance type="ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD" priority="                                                   ="1">
  <Class classname="SimpleWebServer">
    <SourceLine classname="SimpleWebServer" sourcefile="SimpleWebServe
  </Class>
  <Method classname="SimpleWebServer" name="<init>" signature="()V" is
    <SourceLine classname="SimpleWebServer" start="27" end="29" startB                                               WebServer.java" sourcepath="SimpleWebServer.java"/>
  </Method>
  <Field classname="SimpleWebServer" name="dServerSocket" signature="L
    <SourceLine classname="SimpleWebServer" sourcefile="SimpleWebServe
  </Field>
  <SourceLine classname="SimpleWebServer" start="28" end="28" startBytecode="14" endBytecode="14" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  <SourceLine classname="SimpleWebServer" start="28" end="28" startBytecode="14" endBytecode="14" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
</BugInstance>
```

These are the bugs that are found using spotbugs



```
SimpleWebServer (2)
  Of Concern (2)
    High confidence (1)
      Write to static field from instance method (1)
        Write to static field SimpleWebServer.dServerSocket from instance method new SimpleWebServer() [Of Concern(15), High confidence]
    Normal confidence (1)
```

Below is the information on the bugs and where the bug exists in the program



```
        /* parse the HTTP request */
        StringTokenizer st =
            new StringTokenizer (request, " ");

        command = st.nextToken();
        pathname = st.nextToken();
```

SimpleWebServer.java: 28

Navigation

Write to static field SimpleWebServer.dServerSocket from instance method new SimpleWebServer()
Field SimpleWebServer.dServerSocket

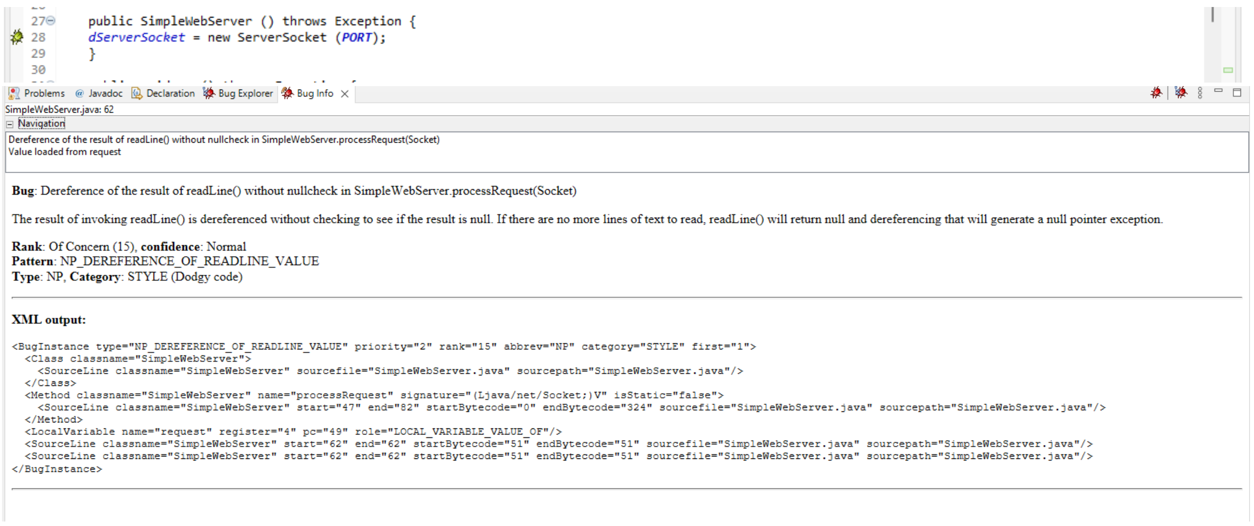
Bug: Write to static field SimpleWebServer.dServerSocket from instance method new SimpleWebServer()

This instance method writes to a static field. This is tricky to get correct if multiple instances are being manipulated, and generally bad practice.

Rank: Of Concern (15), **confidence**: High
Pattern: ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD
Type: ST, **Category**: STYLE (Dodgy code)

XML output:

```
<BugInstance type="ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD" priority="1" rank="15" abbrev="ST" category="STYLE" first="1">
  <Class classname="SimpleWebServer">
    <SourceLine classname="SimpleWebServer" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  </Class>
  <Method classname="SimpleWebServer" name="<init>" signature="()V" isStatic="false">
    <SourceLine classname="SimpleWebServer" start="27" end="29" startBytecode="0" endBytecode="67" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  </Method>
  <Field classname="SimpleWebServer" name="dServerSocket" signature="Ljava/net/ServerSocket;" isStatic="true">
    <SourceLine classname="SimpleWebServer" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  </Field>
  <SourceLine classname="SimpleWebServer" start="28" end="28" startBytecode="14" endBytecode="14" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  <SourceLine classname="SimpleWebServer" start="28" end="28" startBytecode="14" endBytecode="14" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
</BugInstance>
```



```
public SimpleWebServer () throws Exception {
dServerSocket = new ServerSocket (PORT);
}
```

SimpleWebServer.java: 62

Dereference of the result of readLine() without nullcheck in SimpleWebServer.processRequest(Socket)
Value loaded from request

Bug: Dereference of the result of readLine() without nullcheck in SimpleWebServer.processRequest(Socket)

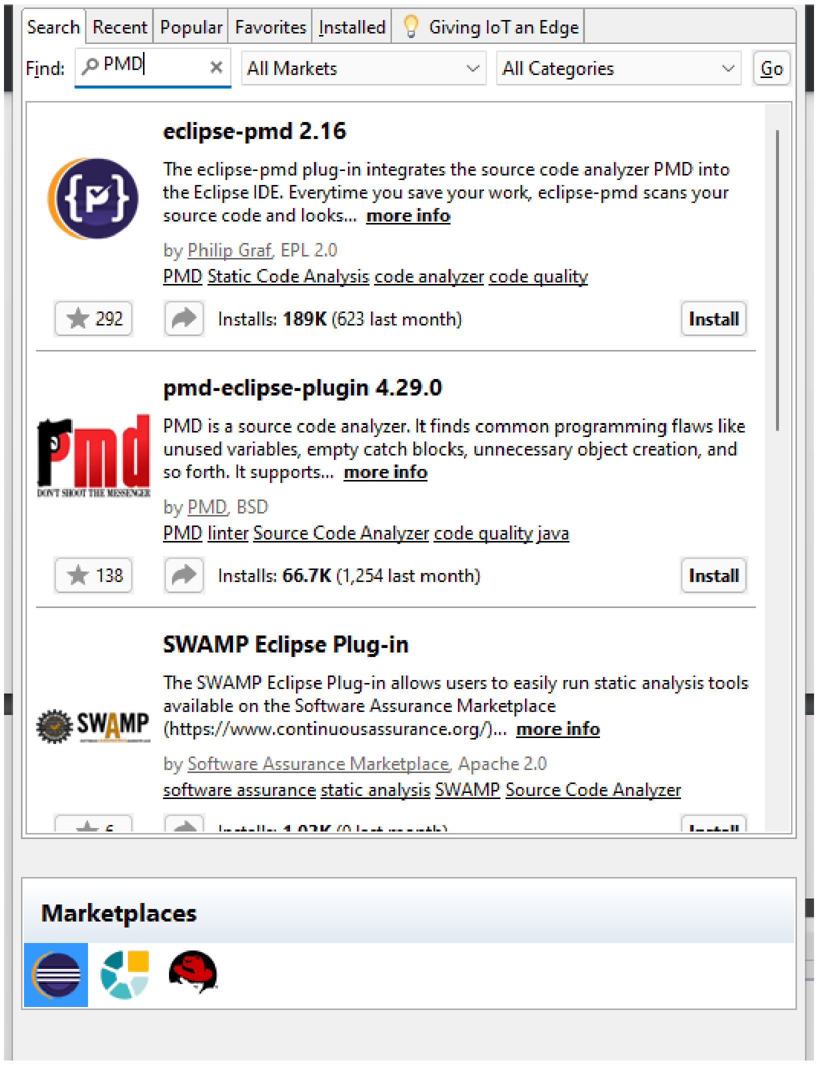
The result of invoking readLine() is dereferenced without checking to see if the result is null. If there are no more lines of text to read, readLine() will return null and dereferencing that will generate a null pointer exception.

Rank: Of Concern (15), **confidence**: Normal
Pattern: NP_DEREFERENCE_OF_READLINE_VALUE
Type: NP, **Category**: STYLE (Dodgy code)

XML output:

```
<BugInstance type="NP_DEREFERENCE_OF_READLINE_VALUE" priority="2" rank="15" abbrev="NP" category="STYLE" first="1">
  <Class classname="SimpleWebServer">
    <SourceLine classname="SimpleWebServer" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  </Class>
  <Method classname="SimpleWebServer" name="processRequest" signature="(Ljava/net/Socket;)V" isStatic="false">
    <SourceLine classname="SimpleWebServer" start="47" end="82" startBytecode="0" endBytecode="324" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  </Method>
  <LocalVariable name="request" register="4" pc="49" role="LOCAL_VARIABLE_VALUE_OF"/>
  <SourceLine classname="SimpleWebServer" start="62" end="62" startBytecode="51" endBytecode="51" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
  <SourceLine classname="SimpleWebServer" start="62" end="62" startBytecode="51" endBytecode="51" sourcefile="SimpleWebServer.java" sourcepath="SimpleWebServer.java"/>
</BugInstance>
```

For pmd using the same process install pmd eclipse plugin 4.29.0 and then restart



eclipse-pmd 2.16

The eclipse-pmd plug-in integrates the source code analyzer PMD into the Eclipse IDE. Everytime you save your work, eclipse-pmd scans your source code and looks... **more info**

by Philip Graf, EPL 2.0
PMD Static Code Analysis code analyzer code quality

292 Installs: **189K** (623 last month) **Install**

pmd-eclipse-plugin 4.29.0

PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It supports... **more info**

by PMD, BSD
PMD linter Source Code Analyzer code quality java

138 Installs: **66.7K** (1,254 last month) **Install**

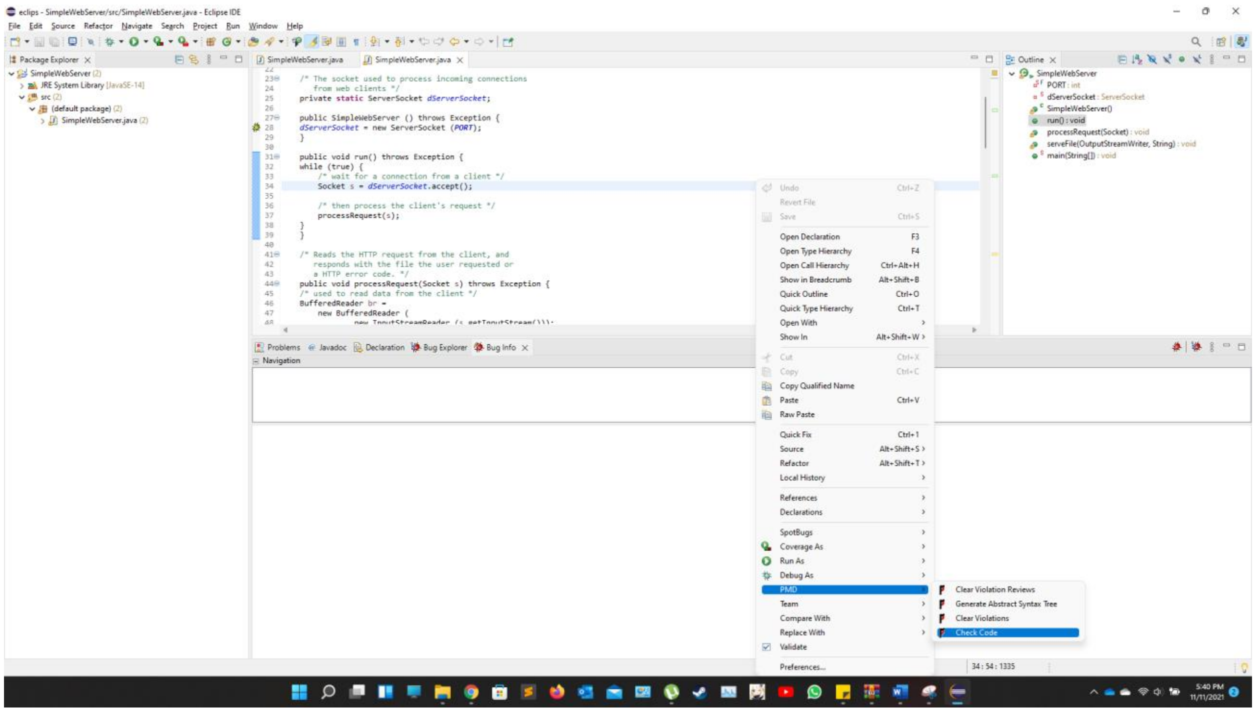
SWAMP Eclipse Plug-in

The SWAMP Eclipse Plug-in allows users to easily run static analysis tools available on the Software Assurance Marketplace (https://www.continuousassurance.org/)... **more info**

by Software Assurance Marketplace, Apache 2.0
software assurance static analysis SWAMP Source Code Analyzer

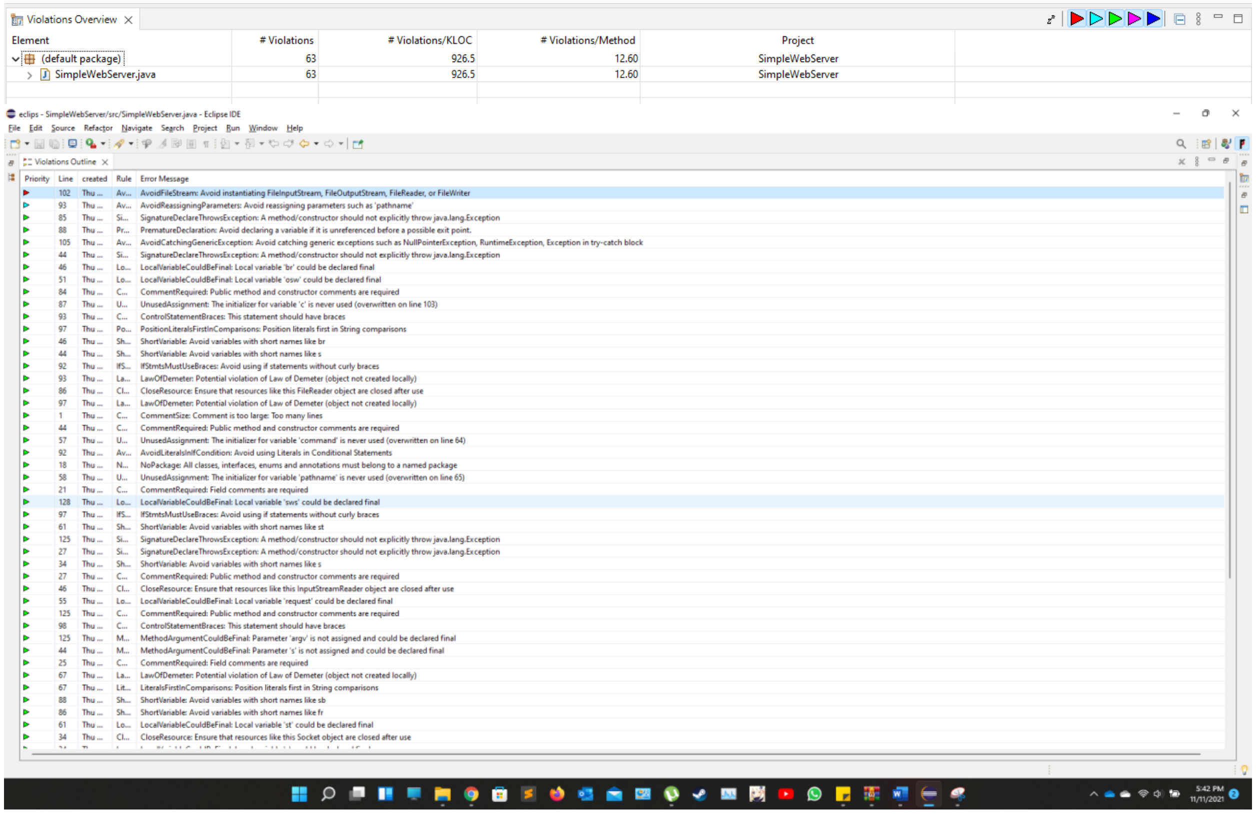
Marketplaces

Once the pmd is installed we can rightclick and go to pmd and click on check code



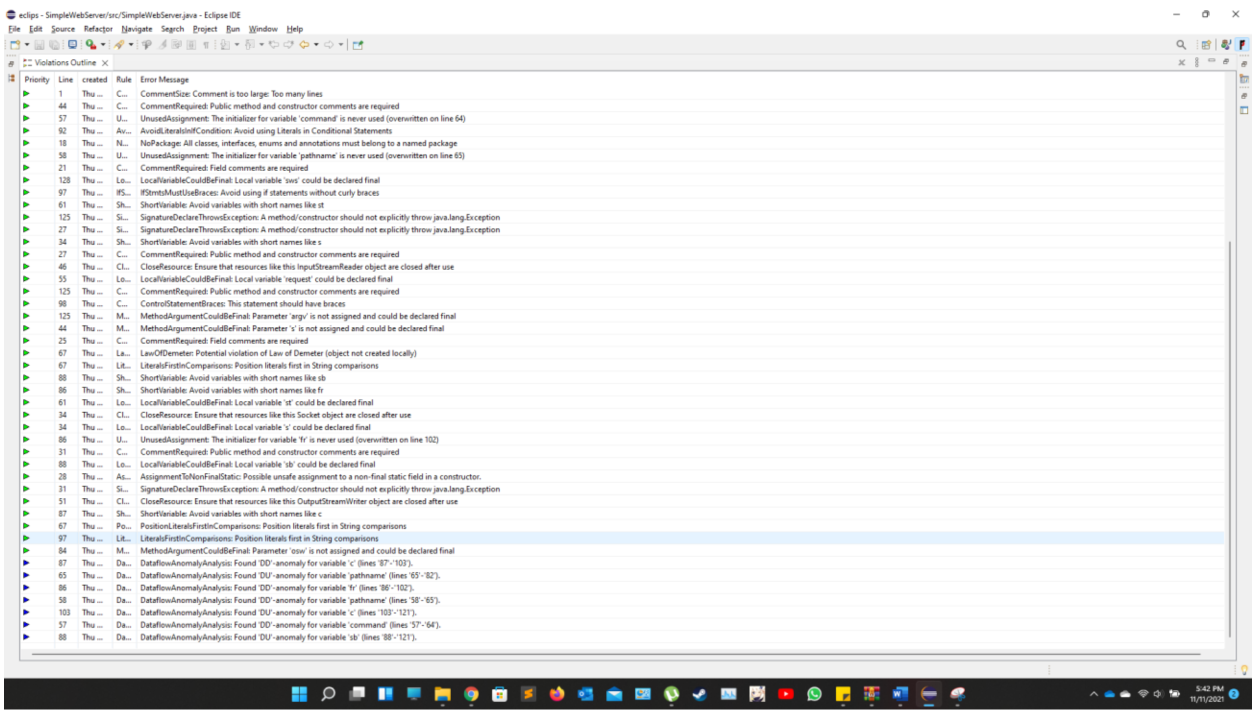
```java
    /* The socket used to process incoming connections
       from web clients */
    private static ServerSocket dServerSocket;

    public SimpleWebServer () throws Exception {
    dServerSocket = new ServerSocket (PORT);
    }

    public void run() throws Exception {
    while (true) {
        /* wait for a connection from a client */
        Socket s = dServerSocket.accept();

        /* then process the client's request */
        processRequest(s);
    }
    }

    /* Reads the HTTP request from the client, and
       responds with the file the user requested or
       a HTTP error code. */
    public void processRequest(Socket s) throws Exception {
    /* used to read data from the client */
    BufferedReader br =
        new BufferedReader (
```

Below is the generated report for the pmd check


Violations Overview
eclips - SimpleWebServer/src/SimpleWebServer.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Violations Outline
5:42 PM
11/11/2021

Element	# Violations	# Violations/KLOC	# Violations/Method	Project
(default package)	63	926.5	12.60	SimpleWebServer
SimpleWebServer.java	63	926.5	12.60	SimpleWebServer

Priority	Line	created	Rule	Error Message
▶	102	Thu ...	Av...	AvoidFileStream: Avoid instantiating FileInputStream, FileOutputStream, FileReader, or FileWriter
▶	93	Thu ...	Av...	AvoidReassigningParameters: Avoid reassigning parameters such as 'pathname'
▶	85	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	88	Thu ...	Pr...	PrematureDeclaration: Avoid declaring a variable if it is unreferenced before a possible exit point.
▶	105	Thu ...	Av...	AvoidCatchingGenericException: Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception in try-catch block
▶	44	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	46	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'br' could be declared final
▶	51	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'osw' could be declared final
▶	84	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	87	Thu ...	U...	UnusedAssignment: The initializer for variable 'c' is never used (overwritten on line 103)
▶	93	Thu ...	C...	ControlStatementBraces: This statement should have braces
▶	97	Thu ...	Po...	PositionLiteralsFirstInComparisons: Position literals first in String comparisons
▶	46	Thu ...	Sh...	ShortVariable: Avoid variables with short names like br
▶	44	Thu ...	Sh...	ShortVariable: Avoid variables with short names like s
▶	92	Thu ...	IfS...	IfStmtsMustUseBraces: Avoid using if statements without curly braces
▶	93	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	86	Thu ...	Cl...	CloseResource: Ensure that resources like this FileReader object are closed after use
▶	97	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	1	Thu ...	C...	CommentSize: Comment is too large: Too many lines
▶	44	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	57	Thu ...	U...	UnusedAssignment: The initializer for variable 'command' is never used (overwritten on line 64)
▶	92	Thu ...	Av...	AvoidLiteralsInIfCondition: Avoid using Literals in Conditional Statements
▶	18	Thu ...	N...	NoPackage: All classes, interfaces, enums and annotations must belong to a named package
▶	58	Thu ...	U...	UnusedAssignment: The initializer for variable 'pathname' is never used (overwritten on line 65)
▶	21	Thu ...	C...	CommentRequired: Field comments are required
▶	128	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sws' could be declared final
▶	97	Thu ...	IfS...	IfStmtsMustUseBraces: Avoid using if statements without curly braces
▶	61	Thu ...	Sh...	ShortVariable: Avoid variables with short names like st
▶	125	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	27	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	34	Thu ...	Sh...	ShortVariable: Avoid variables with short names like s
▶	27	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	46	Thu ...	Cl...	CloseResource: Ensure that resources like this InputStreamReader object are closed after use
▶	55	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'request' could be declared final
▶	125	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	98	Thu ...	C...	ControlStatementBraces: This statement should have braces
▶	125	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'argv' is not assigned and could be declared final
▶	44	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 's' is not assigned and could be declared final
▶	25	Thu ...	C...	CommentRequired: Field comments are required
▶	67	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	67	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	88	Thu ...	Sh...	ShortVariable: Avoid variables with short names like sb
▶	86	Thu ...	Sh...	ShortVariable: Avoid variables with short names like fr
▶	61	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'st' could be declared final
▶	34	Thu ...	Cl...	CloseResource: Ensure that resources like this Socket object are closed after use



eclips - SimpleWebServer/src/SimpleWebServer.java - Eclipse IDE

Violations Outline

Priority	Line	created	Rule	Error Message
▶	1	Thu ...	C...	CommentSize: Comment is too large: Too many lines
▶	44	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	57	Thu ...	U...	UnusedAssignment: The initializer for variable 'command' is never used (overwritten on line 64)
▶	92	Thu ...	Av...	AvoidLiteralsInIfCondition: Avoid using Literals in Conditional Statements
▶	18	Thu ...	N...	NoPackage: All classes, interfaces, enums and annotations must belong to a named package
▶	58	Thu ...	U...	UnusedAssignment: The initializer for variable 'pathname' is never used (overwritten on line 65)
▶	21	Thu ...	C...	CommentRequired: Field comments are required
▶	128	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sws' could be declared final
▶	97	Thu ...	IfS...	IfStmtsMustUseBraces: Avoid using if statements without curly braces
▶	61	Thu ...	Sh...	ShortVariable: Avoid variables with short names like st
▶	125	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	27	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	34	Thu ...	Sh...	ShortVariable: Avoid variables with short names like s
▶	27	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	46	Thu ...	Cl...	CloseResource: Ensure that resources like this InputStreamReader object are closed after use
▶	55	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'request' could be declared final
▶	125	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	98	Thu ...	C...	ControlStatementBraces: This statement should have braces
▶	125	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'argv' is not assigned and could be declared final
▶	44	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 's' is not assigned and could be declared final
▶	25	Thu ...	C...	CommentRequired: Field comments are required
▶	67	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	67	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	88	Thu ...	Sh...	ShortVariable: Avoid variables with short names like sb
▶	86	Thu ...	Sh...	ShortVariable: Avoid variables with short names like fr
▶	61	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'st' could be declared final
▶	34	Thu ...	Cl...	CloseResource: Ensure that resources like this Socket object are closed after use
▶	34	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 's' could be declared final
▶	86	Thu ...	U...	UnusedAssignment: The initializer for variable 'fr' is never used (overwritten on line 102)
▶	31	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	88	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sb' could be declared final
▶	28	Thu ...	As...	AssignmentToNonFinalStatic: Possible unsafe assignment to a non-final static field in a constructor.
▶	31	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	51	Thu ...	Cl...	CloseResource: Ensure that resources like this OutputStreamWriter object are closed after use
▶	87	Thu ...	Sh...	ShortVariable: Avoid variables with short names like c
▶	67	Thu ...	Po...	PositionLiteralsFirstInComparisons: Position literals first in String comparisons
▶	97	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	84	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'osw' is not assigned and could be declared final
▶	87	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'c' (lines '87'-'103').
▶	65	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'pathname' (lines '65'-'82').
▶	86	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'fr' (lines '86'-'102').
▶	58	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'pathname' (lines '58'-'65').
▶	103	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'c' (lines '103'-'121').
▶	57	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'command' (lines '57'-'64').
▶	88	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'sb' (lines '88'-'121').

Comparison/Contrast Tools:

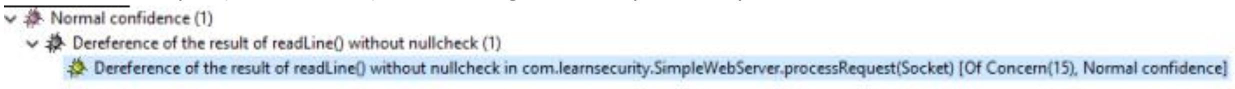
Does the tool analyze source or binary as input?

- PMD used for the analysis of source code. Spotbugs analyzes the binary as input

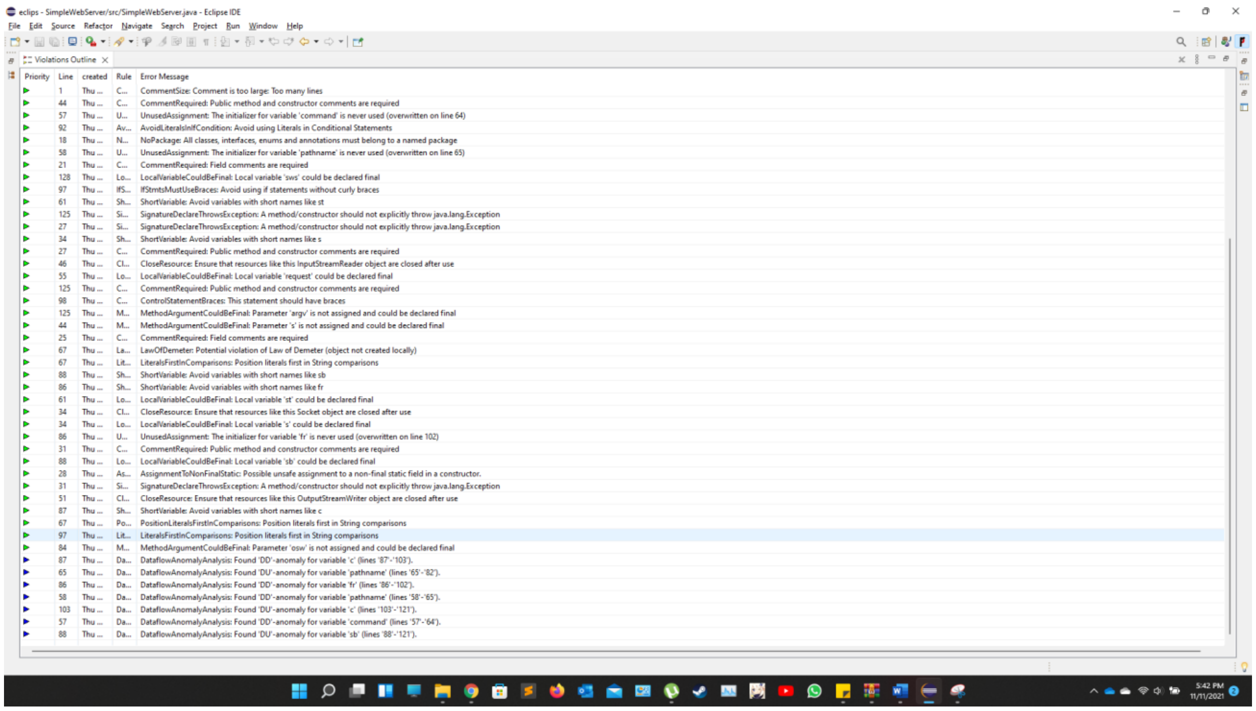
Which category of tools is it?

- Spotbugs is type checking and program verification type
- PMD is style checking and program understanding type

Show an example (if one exists) of a finding that is reported by one tool and not others.



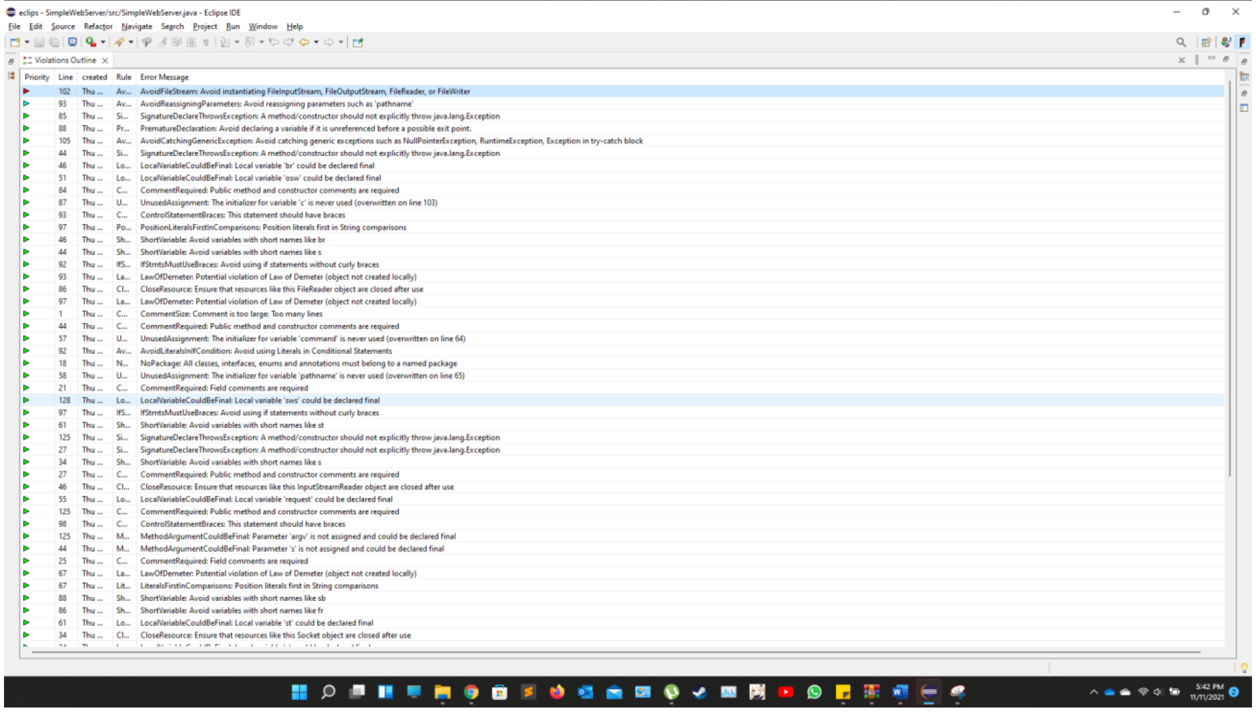
Normal confidence (1)
Dereference of the result of readLine() without nullcheck (1)
Dereference of the result of readLine() without nullcheck in com.learnsecurity.SimpleWebServer.processRequest(Socket) [Of Concern(15), Normal confidence]



eclips - SimpleWebServer/src/SimpleWebServer.java - Eclipse IDE

Violations Outline

Priority	Line	created	Rule	Error Message
▶	1	Thu ...	C...	CommentSize: Comment is too large: Too many lines
▶	44	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	57	Thu ...	U...	UnusedAssignment: The initializer for variable 'command' is never used (overwritten on line 64)
▶	92	Thu ...	Av...	AvoidLiteralsInIfCondition: Avoid using Literals in Conditional Statements
▶	18	Thu ...	N...	NoPackage: All classes, interfaces, enums and annotations must belong to a named package
▶	58	Thu ...	U...	UnusedAssignment: The initializer for variable 'pathname' is never used (overwritten on line 65)
▶	21	Thu ...	C...	CommentRequired: Field comments are required
▶	128	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sws' could be declared final
▶	97	Thu ...	IfS...	IfStmtsMustUseBraces: Avoid using if statements without curly braces
▶	61	Thu ...	Sh...	ShortVariable: Avoid variables with short names like st
▶	125	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	27	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	34	Thu ...	Sh...	ShortVariable: Avoid variables with short names like s
▶	27	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	46	Thu ...	Cl...	CloseResource: Ensure that resources like this InputStreamReader object are closed after use
▶	55	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'request' could be declared final
▶	125	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	98	Thu ...	C...	ControlStatementBraces: This statement should have braces
▶	125	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'argv' is not assigned and could be declared final
▶	44	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 's' is not assigned and could be declared final
▶	25	Thu ...	C...	CommentRequired: Field comments are required
▶	67	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	67	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	88	Thu ...	Sh...	ShortVariable: Avoid variables with short names like sb
▶	86	Thu ...	Sh...	ShortVariable: Avoid variables with short names like fr
▶	61	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'st' could be declared final
▶	34	Thu ...	Cl...	CloseResource: Ensure that resources like this Socket object are closed after use
▶	34	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 's' could be declared final
▶	86	Thu ...	U...	UnusedAssignment: The initializer for variable 'fr' is never used (overwritten on line 102)
▶	31	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	88	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sb' could be declared final
▶	28	Thu ...	As...	AssignmentToNonFinalStatic: Possible unsafe assignment to a non-final static field in a constructor.
▶	31	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	51	Thu ...	Cl...	CloseResource: Ensure that resources like this OutputStreamWriter object are closed after use
▶	87	Thu ...	Sh...	ShortVariable: Avoid variables with short names like c
▶	67	Thu ...	Po...	PositionLiteralsFirstInComparisons: Position literals first in String comparisons
▶	97	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	84	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'osw' is not assigned and could be declared final
▶	87	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'c' (lines '87'-'103').
▶	65	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'pathname' (lines '65'-'82').
▶	86	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'fr' (lines '86'-'102').
▶	58	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'pathname' (lines '58'-'65').
▶	103	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'c' (lines '103'-'121').
▶	57	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'command' (lines '57'-'64').
▶	88	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'sb' (lines '88'-'121').



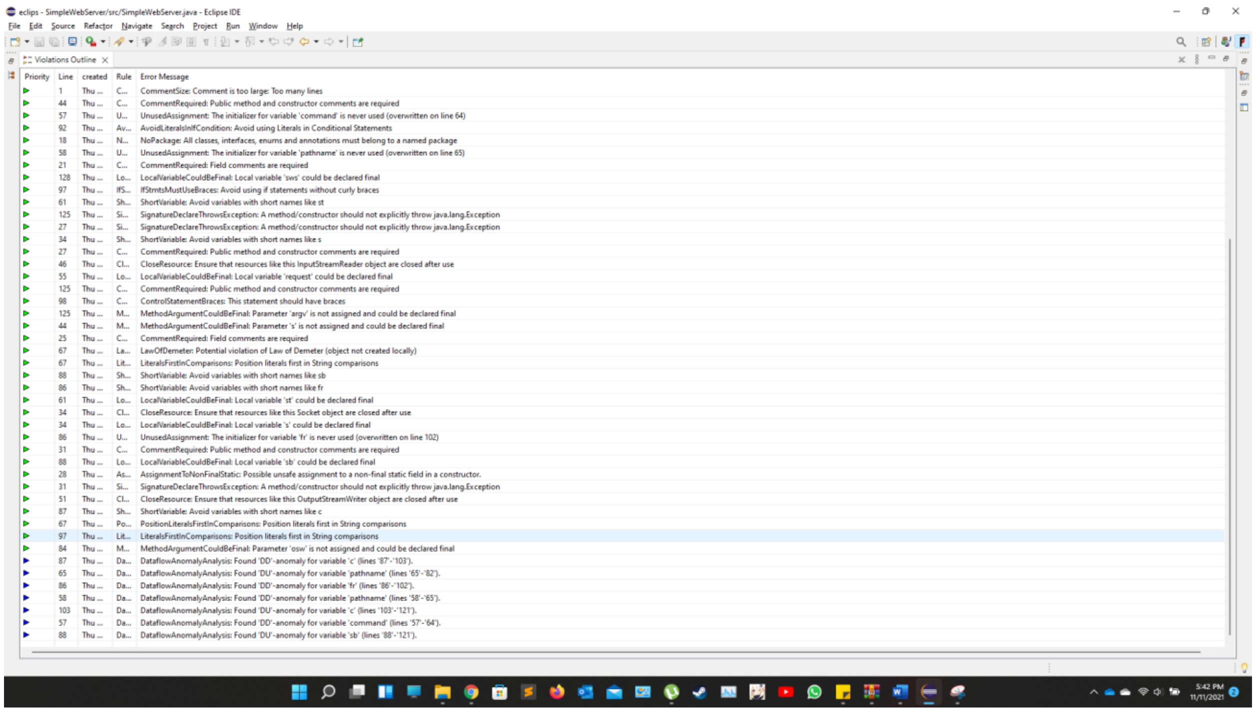
eclips - SimpleWebServer/src/SimpleWebServer.java - Eclipse IDE

Violations Outline

Priority	Line	created	Rule	Error Message
▶	102	Thu ...	Av...	AvoidFileStream: Avoid instantiating FileInputStream, FileOutputStream, FileReader, or FileWriter
▶	93	Thu ...	Av...	AvoidReassigningParameters: Avoid reassigning parameters such as 'pathname'
▶	85	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	88	Thu ...	Pr...	PrematureDeclaration: Avoid declaring a variable if it is unreferenced before a possible exit point.
▶	105	Thu ...	Av...	AvoidCatchingGenericException: Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception in try-catch block
▶	44	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	46	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'br' could be declared final
▶	51	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'osw' could be declared final
▶	84	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	87	Thu ...	U...	UnusedAssignment: The initializer for variable 'c' is never used (overwritten on line 103)
▶	93	Thu ...	C...	ControlStatementBraces: This statement should have braces
▶	97	Thu ...	Po...	PositionLiteralsFirstInComparisons: Position literals first in String comparisons
▶	46	Thu ...	Sh...	ShortVariable: Avoid variables with short names like br
▶	44	Thu ...	Sh...	ShortVariable: Avoid variables with short names like s
▶	92	Thu ...	IfS...	IfStmtsMustUseBraces: Avoid using if statements without curly braces
▶	93	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	86	Thu ...	Cl...	CloseResource: Ensure that resources like this FileReader object are closed after use
▶	97	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	1	Thu ...	C...	CommentSize: Comment is too large: Too many lines
▶	44	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	57	Thu ...	U...	UnusedAssignment: The initializer for variable 'command' is never used (overwritten on line 64)
▶	92	Thu ...	Av...	AvoidLiteralsInIfCondition: Avoid using Literals in Conditional Statements
▶	18	Thu ...	N...	NoPackage: All classes, interfaces, enums and annotations must belong to a named package
▶	58	Thu ...	U...	UnusedAssignment: The initializer for variable 'pathname' is never used (overwritten on line 65)
▶	21	Thu ...	C...	CommentRequired: Field comments are required
▶	128	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sws' could be declared final
▶	97	Thu ...	IfS...	IfStmtsMustUseBraces: Avoid using if statements without curly braces
▶	61	Thu ...	Sh...	ShortVariable: Avoid variables with short names like st
▶	125	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	27	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	34	Thu ...	Sh...	ShortVariable: Avoid variables with short names like s
▶	27	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	46	Thu ...	Cl...	CloseResource: Ensure that resources like this InputStreamReader object are closed after use
▶	55	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'request' could be declared final
▶	125	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	98	Thu ...	C...	ControlStatementBraces: This statement should have braces
▶	125	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'argv' is not assigned and could be declared final
▶	44	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 's' is not assigned and could be declared final
▶	25	Thu ...	C...	CommentRequired: Field comments are required
▶	67	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	67	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	88	Thu ...	Sh...	ShortVariable: Avoid variables with short names like sb
▶	86	Thu ...	Sh...	ShortVariable: Avoid variables with short names like fr
▶	61	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'st' could be declared final
▶	34	Thu ...	Cl...	CloseResource: Ensure that resources like this Socket object are closed after use

We cans see that the bugs reported in spotBugs on line 63 is not there ins PMD report

Show an example (if one exists) of a finding reported by multiple tools.


eclips - SimpleWebServer/src/SimpleWebServer.java - Eclipse IDE

Violations Outline

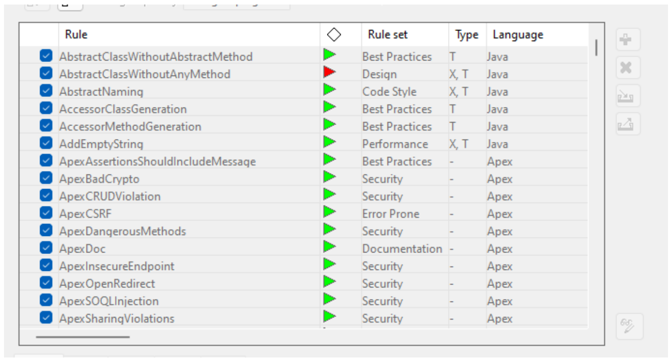
Priority	Line	created	Rule	Error Message
▶	1	Thu ...	C...	CommentSize: Comment is too large: Too many lines
▶	44	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	57	Thu ...	U...	UnusedAssignment: The initializer for variable 'command' is never used (overwritten on line 64)
▶	92	Thu ...	Av...	AvoidLiteralsInIfCondition: Avoid using Literals in Conditional Statements
▶	18	Thu ...	N...	NoPackage: All classes, interfaces, enums and annotations must belong to a named package
▶	58	Thu ...	U...	UnusedAssignment: The initializer for variable 'pathname' is never used (overwritten on line 65)
▶	21	Thu ...	C...	CommentRequired: Field comments are required
▶	128	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sws' could be declared final
▶	97	Thu ...	IfS...	IfStmtsMustUseBraces: Avoid using if statements without curly braces
▶	61	Thu ...	Sh...	ShortVariable: Avoid variables with short names like st
▶	125	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	27	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	34	Thu ...	Sh...	ShortVariable: Avoid variables with short names like s
▶	27	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	46	Thu ...	Cl...	CloseResource: Ensure that resources like this InputStreamReader object are closed after use
▶	55	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'request' could be declared final
▶	125	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	98	Thu ...	C...	ControlStatementBraces: This statement should have braces
▶	125	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'argv' is not assigned and could be declared final
▶	44	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 's' is not assigned and could be declared final
▶	25	Thu ...	C...	CommentRequired: Field comments are required
▶	67	Thu ...	La...	LawOfDemeter: Potential violation of Law of Demeter (object not created locally)
▶	67	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	88	Thu ...	Sh...	ShortVariable: Avoid variables with short names like sb
▶	86	Thu ...	Sh...	ShortVariable: Avoid variables with short names like fr
▶	61	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'st' could be declared final
▶	34	Thu ...	Cl...	CloseResource: Ensure that resources like this Socket object are closed after use
▶	34	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 's' could be declared final
▶	86	Thu ...	U...	UnusedAssignment: The initializer for variable 'fr' is never used (overwritten on line 102)
▶	31	Thu ...	C...	CommentRequired: Public method and constructor comments are required
▶	88	Thu ...	Lo...	LocalVariableCouldBeFinal: Local variable 'sb' could be declared final
▶	28	Thu ...	As...	AssignmentToNonFinalStatic: Possible unsafe assignment to a non-final static field in a constructor.
▶	31	Thu ...	Si...	SignatureDeclareThrowsException: A method/constructor should not explicitly throw java.lang.Exception
▶	51	Thu ...	Cl...	CloseResource: Ensure that resources like this OutputStreamWriter object are closed after use
▶	87	Thu ...	Sh...	ShortVariable: Avoid variables with short names like c
▶	67	Thu ...	Po...	PositionLiteralsFirstInComparisons: Position literals first in String comparisons
▶	97	Thu ...	Lit...	LiteralsFirstInComparisons: Position literals first in String comparisons
▶	84	Thu ...	M...	MethodArgumentCouldBeFinal: Parameter 'osw' is not assigned and could be declared final
▶	87	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'c' (lines '87'-'103').
▶	65	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'pathname' (lines '65'-'82').
▶	86	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'fr' (lines '86'-'102').
▶	58	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'pathname' (lines '58'-'65').
▶	103	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'c' (lines '103'-'121').
▶	57	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DD'-anomaly for variable 'command' (lines '57'-'64').
▶	88	Thu ...	Da...	DataflowAnomalyAnalysis: Found 'DU'-anomaly for variable 'sb' (lines '88'-'121').

The bug in lone 29 is shown in both the reports

For the known flaw in the code used, document which tools reported it (true negative) and which tools did not (false positive).

- PMD reports way more bugs when compared to Spotbugs. An example would be, PMD shows that the socket is not closed in the code which is a flaw. If the socket is bound to any of the addresses then the address is permanently lost as the socket is not closed.

Result:

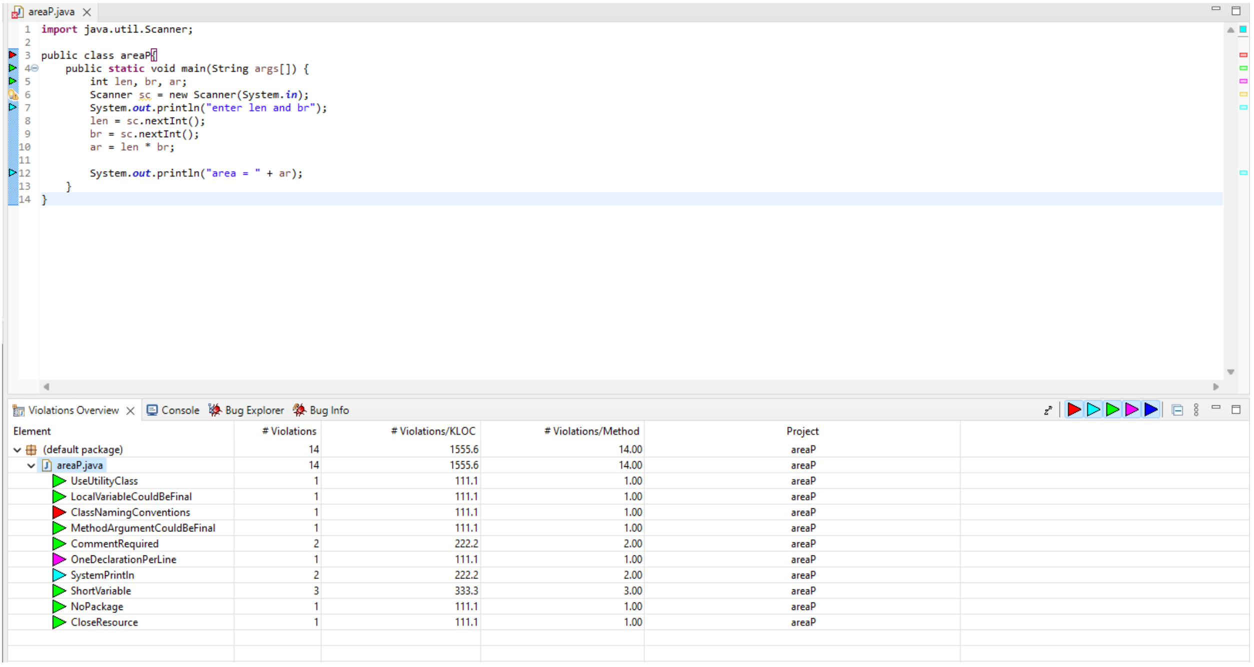


	Rule	◇	Rule set	Type	Language
☑	AbstractClassWithoutAbstractMethod	▶	Best Practices	T	Java
☑	AbstractClassWithoutAnyMethod	▶	Design	X, T	Java
☑	AbstractNaming	▶	Code Style	X, T	Java
☑	AccessorClassGeneration	▶	Best Practices	T	Java
☑	AccessorMethodGeneration	▶	Best Practices	T	Java
☑	AddEmptyString	▶	Performance	X, T	Java
☑	ApexAssertionsShouldIncludeMessage	▶	Best Practices	-	Apex
☑	ApexBadCrypto	▶	Security	-	Apex
☑	ApexCRUDViolation	▶	Security	-	Apex
☑	ApexCSRF	▶	Error Prone	-	Apex
☑	ApexDangerousMethods	▶	Security	-	Apex
☑	ApexDoc	▶	Documentation	-	Apex
☑	ApexInsecureEndpoint	▶	Security	-	Apex
☑	ApexOpenRedirect	▶	Security	-	Apex
☑	ApexSOQLInjection	▶	Security	-	Apex
☑	ApexSharingViolations	▶	Security	-	Apex

We use the most aggressive approach that is possible

PART 2:

This is a simple code for area of a rectange



areaP.java

```java
import java.util.Scanner;

public class areaP{
    public static void main(String args[]) {
        int len, br, ar;
        Scanner sc = new Scanner(System.in);
        System.out.println("enter len and br");
        len = sc.nextInt();
        br = sc.nextInt();
        ar = len * br;

        System.out.println("area = " + ar);
    }
}
```

Violations Overview | Console | Bug Explorer | Bug Info

Element	# Violations	# Violations/KLOC	# Violations/Method	Project
(default package)	14	1555.6	14.00	areaP
areaP.java	14	1555.6	14.00	areaP
UseUtilityClass	1	111.1	1.00	areaP
LocalVariableCouldBeFinal	1	111.1	1.00	areaP
ClassNamingConventions	1	111.1	1.00	areaP
MethodArgumentCouldBeFinal	1	111.1	1.00	areaP
CommentRequired	2	222.2	2.00	areaP
OneDeclarationPerLine	1	111.1	1.00	areaP
SystemPrintln	2	222.2	2.00	areaP
ShortVariable	3	333.3	3.00	areaP
NoPackage	1	111.1	1.00	areaP
CloseResource	1	111.1	1.00	areaP

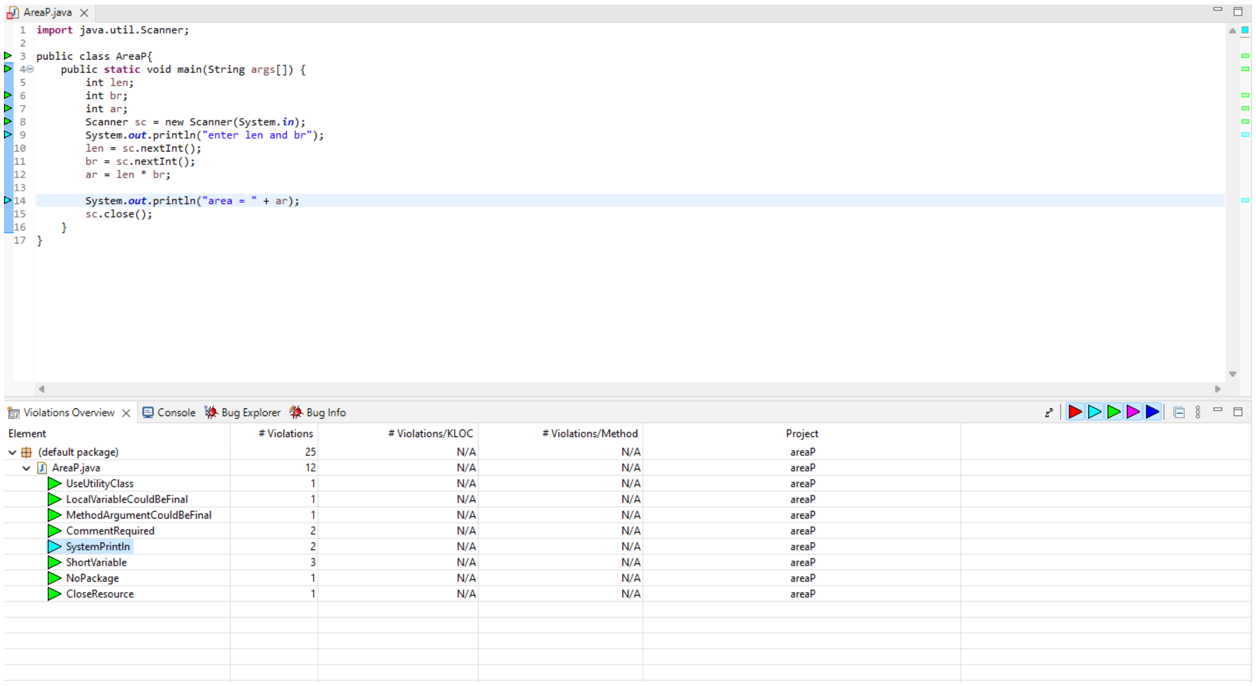
The scanner classs is not closed in this program which can cause errors.
By manual analysis we close the scanner class in the next part of the task

Result:
Doing the sportbugs check we don't find any bugs in the code and then when we use the PMD check we have found the violations which are showed in the above image

We fixed the code that is causing the violations In the before image and have cleared most of the violations that have occurred

1. We declare variables in different lines
2. The useless parentheses violation is auto fixed while clearing the other bugs

The image after fixing the bugs is shown below



```java
import java.util.Scanner;

public class AreaP{
    public static void main(String args[]) {
        int len;
        int br;
        int ar;
        Scanner sc = new Scanner(System.in);
        System.out.println("enter len and br");
        len = sc.nextInt();
        br = sc.nextInt();
        ar = len * br;

        System.out.println("area = " + ar);
        sc.close();
    }
}
```

Element	# Violations	# Violations/KLOC	# Violations/Method	Project
(default package)	25	N/A	N/A	areaP
AreaP.java	12	N/A	N/A	areaP
UseUtilityClass	1	N/A	N/A	areaP
LocalVariableCouldBeFinal	1	N/A	N/A	areaP
MethodArgumentCouldBeFinal	1	N/A	N/A	areaP
CommentRequired	2	N/A	N/A	areaP
SystemPrintln	2	N/A	N/A	areaP
ShortVariable	3	N/A	N/A	areaP
NoPackage	1	N/A	N/A	areaP
CloseResource	1	N/A	N/A	areaP